



660220V da

MRC Technical Summary Report #1968 FFT AS NESTED MULTIPLICATION, WITH A TWIST Carl de Boor

Mathematics Research Center University of Wisconsin–Madison 610 Walnut Street Madison, Wisconsin 53706

June 1979 HR Received May 11, 1979 DECORPTION

Approved for public release Distribution unlimited

Sponsored by

U. S. Army Research Office P.O. Box 12211 Research Triangle Park North Carolina 27709

UNIVERSITY OF WISCONSIN - MADISON MATHEMATICS RESEARCH CENTER

FFT AS NESTED MULTIPLICATION, WITH A TWIST

Carl de Boor

Technical Summary Report #1968 June 1979

ABSTRACT

A simple, yet complete and detailed description of the Fast Fourier Transform for general N is given, with the aim of making the underlying idea quite apparent. To help with this didactic goal, a simple twist, i.e., a shifting of information from rows to columns during the calculations, is introduced which allows to give a simple meaning to intermediate results and assures that the final results need no further reordering.

AMS(MOS) Subject Classification: 65T05

Key Words: FFT = Fast Fourier Transform nested multiplication

Work Unit No. 8 - Computer Science

Acces	ssion For	1
NTIS DDC 1	GRA&I AB	R
Unann	ounced	HI
Justi	fication	L I
Ey		
Distr	ibut ion/	
_Avei	ability C	edes
Dist	Avail and, special	/or
a		
III		S. Salar

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

A. -A-

Significance and Explanation

In the last twenty years, various forms of a fast Fourier transform (FFT) have become popular. The various algorithms have in common that they produce the discrete Fourier transform on N points in about N $\ln(N)$ rather than N^2 operations. But, while these ideas, notably through Cooley & Tukey, have found wide application in computations, their didactic treatment has left something to be desired.

This report is an attempt to remedy this situation.

FFT AS NESTED MULTIPLICATION, WITH A TWIST

Carl de Boor

1. Introduction. The discrete Fourier transform (DFT) $\hat{z} = F_N z$ of an N-vector \underline{z} is given by the rule (1) $\hat{z}_v = \sum_{n=1}^N z_n \omega_N^{(v-1)(n-1)}$, v = 1, ..., N,

with

(2)
$$\omega_{-} = \exp(-\sqrt{-1} 2\pi/N)$$

a principal N-th root of unity. Thus, \hat{z}_{v} is given as the value of a polynomial of degree < N at the point ω_{N}^{v-1} and can therefore be calculated, by nested multiplication, in N operations. Here, I follow Cooley & Tukey [1] in counting a complex multiplication followed by a complex addition as one <u>operation</u>.

In the last twenty years, various forms of a fast Fourier transform (FFT) have become popular. The various algorithms have in common that they produce the DFT on N points in about N $\ln(N)$ rather than N^2 operations. See Winograd [10] for the latest developments. But, while these ideas, notably through Cooley & Tukey [1], have found wide application in computations, their didactic treatment has left something to be desired.

In a recent article [7], H. R. Schwarz attempts, as he says, to remove the mystical aspect which the FFT has for many people. He does this by describing the FFT in terms of a factorization of the transformation matrix, an idea which he ascribes to Theilheimer [9] but which occurs already in Good [5] where a FFT different from that of Cooley & Tukey is given. A factorization of the transformation matrix is also the basic idea on which Glassman [4] builds his FFT, and Drubin [2] has refined this further; see Ferguson [3] for a lucid description and a simple Fortran program.

By contrast, I want to give here what I believe to be a simple description of the FFT for a general N in terms of <u>nested multiplication</u>. Certainly, Cooley & Tukey [1] thought of the FFT in these terms.

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

2. The case of two factors. Suppose that N = PQ for two integers P and Q greater than 1. Think of the N-vector \underline{z} as stored Fortran fashion in a one-dimensional array. Then we can interpret that array also Fortran fashion as a two-dimensional array Z, of dimension (P,Q). This means that (3) $Z(p,q) = \underline{z}_{p+P(q-1)}$, $p = 1, \dots, P$, $q = 1, \dots, Q$.

Correspondingly, factor the sum (1) for \hat{z}_{y} into a double sum,

$$\hat{z}_{v} = \sum_{p=1}^{P} \sum_{q=1}^{Q} z(p,q) \ \omega_{N}^{(v-1)[p-1 + P(q-1)]}$$
$$= \sum_{p=1}^{P} \left[\sum_{q=1}^{Q} z(p,q) \ \omega_{Q}^{(v-1)(q-1)} \right] \ \omega_{N}^{(v-1)(p-1)}$$

Here, we have made use of the fact that

 $\omega_{\rm P}^{\rm P} = \omega_{\rm Q}$.

This makes apparent the crucial fact that <u>the inner sum</u> in the last right hand side is Q-periodic in v, i.e., replacing v by v+Q does not change its value, due to the fact that $\omega_Q^Q = 1$. This means that we need only calculate this sum for $v = 1, \ldots, Q$ (and for each p). Thus, for each $p = 1, \ldots, P$, we calculate from the Q-vector Z(p, \cdot) the Q-vector whose entries are the numbers

(4) $\sum_{q=1}^{Q} Z(p,q) \omega_{Q}^{(\nu-1)(q-1)}, \quad \nu = 1,...,Q,$

i.e., we calculate the DFT $F_Q^Z(p, \cdot)$, p = 1, ..., P, at a total cost of $P \cdot Q^2 = N \cdot Q$ operations.

Now, we could store the transform of $Z(p, \cdot)$ over $Z(p, \cdot)$. But in anticipation of further developments, we choose to store the transform $F_Q^{Z}(p, \cdot)$ in $Z_1(\cdot,p)$, where Z_1 is a two dimensional array of size (Q,P), rather than (P,Q).

With this, the calculation of $\hat{z}_{_{U}}$ is reduced to the evaluation of the sum

(5)
$$\hat{z}_{v} = \sum_{p=1}^{p} z_{1}(v_{Q}, p) \omega_{N}^{(v-1)(p-1)}, v = 1, ..., N$$

Here, we have used the notation v_Q to indicate the integer between 1 and Qfor which $v - v_Q$ is divisible by Q. At this point, it becomes convenient to think of the one-dimensional array which is to contain the N-vector \hat{z} equivalently as a two-dimensional array Z_0 , of size (Q,P). This means that

(6) $\hat{z}_{\nu + Q(\mu-1)} = Z_0(\nu,\mu)$, for $\nu = 1,...,Q$, $\mu = 1,...,P$. With this, (5) can be written equivalently as

$$z_{0}(v,\mu) = \sum_{p=1}^{P} z_{1}(v,p) \omega_{N}^{[v-1 + Q(\mu-1)](p-1)}$$

 $v = 1, ..., 0, \mu = 1, ..., P$.

Here, the right hand side is a polynomial of degree < P in the quantity $\omega_N^{\nu-1} + Q(\mu-1)$. This quantity can be generated as one goes along, as in the following convenient arrangement of the calculations:

x := 1 for $\mu = 1, ..., P$, do: for $\nu = 1, ..., Q$, do: $Z_0(\nu, \mu) := \sum_{p=1}^{P} Z_1(\nu, p) x^{p-1}$ x := x · ω_N .

The sum in the innermost loop is, of course, to be evaluated by nested multiplication. The total cost of this step is then $Q \cdot P^2 = N \cdot P$ operations (if we neglect the N multiplications needed to generate the various x's). In this way, we have obtained in Z_0 the discrete Fourier transform \hat{z} of z at a cost of only N(P+Q) rather than N² operations.

(7)

3. The general case. "It is easy to see how successive applications of the above procedure, starting with its application to (4), give an m-step algorithm requiring

 $T = N(P_1 + P_2 + ... + P_m)$

operations, where

(8)

$$N = P_1 \cdot P_2 \cdot \ldots \cdot P_m \cdot "$$

So say Cooley & Tukey [1] (except for a change in symbols and equation numbers). In effect, they point out that the first step of the calculations above consists in forming the DFT of various Q-vectors. Hence if Q itself is the product of two integers greater than 1, this calculation can be carried out in fewer than Q^2 operations by applying the same procedure to it, etc. The actual implementation of this idea may not be immediately obvious, though. For this reason, I now discuss a slightly different (and novel) view, according to which the entire transform can be effected by m applications of a slightly enlarged version of (7).

The basic idea is to interpret the storage arrays for the various N-vectors involved in various ways as multidimensional arrays and to shift information appropriately from "rows to columns" as we did earlier when storing the DFT of the row $Z(p, \cdot)$ of Z in the column $Z_1(\cdot, p)$ of Z_1 . For this, I need some notation to indicate that a given one-dimensional array is being considered equivalently as a two- or three-dimensional array.

If Z is a one-dimensional array of length N, then z^A denotes the equivalent two-dimensional array of dimension (A,N/A), and $z^{A,B}$ denotes the equivalent three-dimensional array of dimension (A,B,N/(AB)). Thus

(9) $Z^{A,B}(a,b,c) = Z^{A}(a,b+B(c-1)) = Z^{AB}(a+A(b-1),c)$ = Z(a+A(b-1+B(c-1))).

Let now Z be a one-dimensional array containing \underline{z} , as before, and, for k = 0, ..., m, let Z_k be a one-dimensional array satisfying (10) $Z_k^A(\cdot, c) = F_A Z^{BP}(c, \cdot), c = 1, ..., BP,$ with

11.5

-4-

 $B := B_k := P_1 \cdot \ldots \cdot P_{k-1}, P := P_k, A := A_k := P_m \cdot \ldots \cdot P_{k+1}$ (11) Then $Z_{m} = Z$, and Z_{0} contains $\hat{z} = F_{N} z$. Further, with A, P, B as given by (11), one obtains Z_{k-1} from Z_k by the following slightly extended version of (7):

x := 1
for p = 1,...,P, do:
for a = 1,...,A, do:
for b = 1,...,B, do:

$$z_{k-1}^{A,P}(a,p,b) := \int_{\pi=1}^{P} z_{k}^{A,B}(a,b,\pi) \cdot x^{\pi-1}$$

x := x · ω_{AP} .

Indeed, the algorithm produces

$$z_{k-1}^{A,P}(a,p,b) = \sum_{\pi=1}^{P} z_{k}^{A,B}(a,b,\pi) \omega_{AP}^{[a-1 + A(p-1)](\pi-1)} .$$

On the other hand, (10) implies that

$$z_{k}^{A,B}(\cdot,b,\pi) = F_{A} z^{B,P}(b,\pi,\cdot) = \sum_{\alpha=1}^{A} z^{B,P}(b,\pi,\alpha) \omega_{A}^{(\cdot-1)(\alpha-1)}$$

any

Therefore,

But 1

integer multiple of AP , and this allows the conclusion that

$$Z_{k-1}^{\mathbf{A},\mathbf{P}}(\mathbf{a},\mathbf{p},\mathbf{b}) = \sum_{\pi=1}^{\mathbf{P}} \sum_{\alpha=1}^{\mathbf{A}} z^{\mathbf{B},\mathbf{P}}(\mathbf{b},\pi,\alpha) \omega_{\mathbf{A}\mathbf{P}}^{[\mathbf{a}-1+\mathbf{A}(\mathbf{p}-1)][\pi-1+\mathbf{P}(\alpha-1)]}$$

and so proves that Z_{k-1} , as produced by (12), satisfies (10) (with k replaced by k-1).

This shows that the DFT \hat{z} is obtainable, in Z_0 , by m applications of algorithm (21), starting from Z_m containing z. Since the k-th such application costs $P_{k}A_{k}B_{k}P_{k} = N \cdot P_{k}$ operations, the total number of required operations is indeed given by (8).

In a Fortran implementation of the algorithm, one would, of course, need only two arrays to play the role, in alternation, of the m+l arrays Z_m, \ldots, Z_0 .

(12)

```
SUBROUTINE FFT ( Z1, Z2, N, INZEE )
CONSTRUCTS THE DISCRETE FOURIER TRANSFORM OF Z1 (OR Z2) IN THE COOLEY-
C
   TUKEY WAY, BUT WITH A TWIST.
       INTEGER INZEE, N,
                            AFTER, BEFORE, NEXT, NEXTMX, NOW, PRIME(12)
      COMPLEX Z1(N), Z2(N)
         INPUT
C*****
   Z1, Z2 COMPLEX N-VECTORS
N LENGTH OF Z1 AND Z2
C
C
   INZEE INTEGER INDICATING WHETHER Z1 OR Z2 IS TO BE TRANSFORMED
C
      = 1 , TRANSFORM
= 2 , TRANSFORM
C
                          Z 1
C
                          22
C = 2 , IRANSFORM 22
C****** WORKAREAS ******
  Z1, Z2 ARE BOTH USED AS WORKARRAYS
C
C****** OUTPUT
                         ******
С
  Z1 OR Z2 CONTAINS THE DESIRED TRANSFORM (IN THE CORRECT ORDER)
   INZEE INTEGER INDICATING WHETHER 21 OR 22 CONTAINS THE TRANSFORM,
C
  = 1, TRANSFORM IS IN Z1
= 2, TRANSFORM IS IN Z2
***** METHOD ******
C
C*
      THE INTEGER N IS DIVIDED INTO ITS PRIME FACTORS (UP TO A POINT).
C
   FOR EACH SUCH FACTOR P, THE P-TRANSFORM OF APPROPRIATE P-SUBVECTORS
OF Z1 (OR Z2) IS CALCULATED IN F F T S T P AND STORED IN A SUIT-
C
C
   ABLE WAY IN Z2 (OR Z1). SEE TEXT FOR DETAILS.
С
С
       DATA NEXTMX, PRIME / 12, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37 /
       AFTER = 1
       BEFORE = N
       NEXT = 1
C
   10 IF ((BEFORE/PRIME(NEXT)) *PRIME(NEXT) .LT. BEFORE) THEN
          NEXT = NEXT + 1
IF (NEXT .LE. NEXTMX) THEN
                                              GO TO 10
          ELSE
              NOW = BEFORE
             BEFORE = 1
          END IF
       ELSE
          NOW = PRIME(NEXT)
          BEFORE = BEFORE/PRIME(NEXT)
       END IF
C
       IF (INZEE .EQ. 1) THEN
CALL FFTSTP( Z1, AFTER, NOW, BEFORE, Z2 )
       ELSE
          CALL FFISIP( Z2, AFTER, NOW, BEFORE, Z1 )
       END IF
       INZEE = 3 - INZEE
       IF (BEFORE .EQ. 1)
                                              RETURN
       AFTER = AFTER*NOW
                                              GO TO 10
      END
```

-6-

SUBROUTINE FFISIP (ZIN, AFTER, NOW, BEFORE, ZOUT) CALLED IN FFT CARRIES OUT ONE STEP OF THE DISCRETE FAST FOURIER TRANSFORM. INTEGER AFTER, BEFORE, NOW, IA, IB, IN, J REAL ANGLE, RATIO, TWOPI COMPLEX ZIN(AFTER, BEFORE, NOW), ZOUT(AFTER, NOW, BEFORE), ARG, OMEGA, VALUE DATA TWOPI / 6.2831 85307 17958 64769 / ANGLE = TWOPI/FLOAT(NOW*AFTER) OMEGA = CMPLX(COS(ANGLE), -SIN(ANGLE)) ARG = CMPLX(1.,0.) DO 100 J=1, NOW DO 90 IA=1, AFTER DO 80 IB=1, BEFORE VALUE = ZIN(IA, IB, NOW) DO 70 IN=NOW-1, 1, -VALUE = VALUE * ARG + ZIN(IA, IB, IN) ZOUT(IA, J, IB) = VALUE 70 80 90 ARG = ARG *OMEGA 100 CONTINUE RETURN END

There is no claim that the above program is competitive with the carefully constructed codes such as that of Singleton [8]. Its virtue lies chiefly in its simplicity and transparency. On the other hand, Eric Grosse [6] found that the above code, modified to give special treatment in FFTSTP for the case NOW = 2, and to avoid subroutine calls for the complex arithmetic operations, and compiled by an optimizing compiler, needed only 1.5 to 2 times as much computing time as did Singleton's program for a variety of choices of N.

Finally, the above discussion is based on the Fortran convention whereby multidimensional arrays are stored "column by column", i.e., with the first index running fastest. It is easy to base the discussion instead on the Algol convention whereby arrays are stored "row by row", i.e., with the last index running fastest.

<u>Acknowledgement</u>. I am grateful to Warren Ferguson for several discussions concerning fast Fourier transforms and for comments on an earlier draft. I am indebted to Eric Grosse for carrying out the comparisons mentioned above and for suggesting that the more leisurely discussion in an earlier draft be replaced by showing directly that the Z_k as generated by (12) satisfy (10).

-7-

REFERENCES

- J. W. Cooley & J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comp. <u>19</u> (1965) 297-301.
- M. Drubin, Kronecker product factorization of the FFT matrix, IEEE Trans. on Computers, <u>C-20</u> (1971) 590-593.
- W. Ferguson, A simple derivation of Glassman's general N fast Fourier transform, Mathematics Research Center Technical Summary Report # (1979).
- J. A. Glassman, A generalization of the fast Fourier transform, IEEE Trans. on Computers, <u>C-19</u> (1970) 105-116.
- J. Good, The interaction algorithm and practical Fourier series, J. Roy. Statist. Soc. Ser. B <u>20</u> (1958) 361-372; Addendum, <u>22</u> (1960) 372-375.
- 6. Eric Grosse, letter dated April 6, 1979.
- H. R. Schwarz, Elementare Darstellung der schnellen Fouriertransformation, Computing <u>19</u> (1977) 107-116.
- R. C. Singleton, Algorithm 339. An Algol procedure for the Fast Fourier Transform with arbitrary factors, Comm. Assoc. Comp. Mach. <u>11</u> (1968) 776ff.
- F. Theilheimer, A matrix version of the Fast Fourier Transform, IEEE Trans. on Audio and Electroacoustics <u>17</u> (1969) 158-161.
- S. Winograd, On computing the discrete Fourier transform, Math. Comp. <u>32</u> (1978) 175-199.

CdB:db

-8-

	TION PAGE	BEFORE COMPLETING FORM
. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
1968		
TITLE (and Subtitie)		5. TYPE OF REPORT & PERIOD COVERED
and for a second s	and the second	Summary Report - no specific
FFT AS NESTED MULTIPLICATI	ON, WITH A TWIST	reporting period
	and a second	6. PERFORMING ORG. REPORT NUMBER
AUTHOR(•)	~	8. CONTRACT OR GRANT NUMBER(8)
10 Carl de Boor	(1)	DAAG29-75-C-0024
PERFORMING ORGANIZATION NAME AND AD	DRESS	10. PROGRAM ELEMENT, PROJECT, TASK
Mathematics Research Center,	University of	AREA & WORK UNIT NUMBERS
610 Walnut Street	Wisconsin	8 - Computer Science
Madison, Wisconsin 53706		
1. CONTROLLING OFFICE NAME AND ADDRES	s	12. REPORT DATE
U. S. Army Research Office		4 June 1979
P.O. Box 12211		13: NUMBER OF PAGES
Research Triangle Park, North	Carolina 27709	8
4. MONITORING AGENCY NAME & ADDRESS(IF	dillerent from Controlling Office)	13. SECURITY CLASS. (of this report)
(14 MRC-TIDD	-1968	UNCLASSIFIED
CULTER DA		15. DECLASSIFICATION DOWNGRADING
		JUNEDULE
7. DISTRIBUTION STATEMENT (of the abstract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the abstract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the abstract	entered in Block 20, il different fro	m Report)
7. DISTRIBUTION STATEMENT (of the abetract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the ebstrect B. SUPPLEMENTARY NOTES	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the abetract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the ebstract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the abstract	entered in Block 20, if different fro	m Report)
7. DISTRIBUTION STATEMENT (of the ebetrect 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse side if neces	entered in Block 20, if different fro neary and identify by block number)	m Report)
7. DISTRIBUTION STATEMENT (of the ebetrect 18. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfo	entered in Block 20, if different fro neary and identify by block number) TM	m Report)
 DISTRIBUTION STATEMENT (of the ebetrect SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse eide II necession FFT = Fast Fourier Transfonested multiplication 	entered in Block 20, if different fro beary and identify by block number) YM	m Report)
 DISTRIBUTION STATEMENT (of the ebstract SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse side if neces FFT = Fast Fourier Transfo nested multiplication 	entered in Block 20, if different fro entered in Block 20, if different fro seary and identify by block number) IM	m Report)
 7. DISTRIBUTION STATEMENT (of the ebstract 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse side if necessing FFT = Fast Fourier Transfore nested multiplication 	entered in Block 20, if different fro neary and identify by block number) YM	m Report)
 DISTRIBUTION STATEMENT (of the ebetract SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfo nested multiplication ABSTRACT (Continue on reverse eide if neces 	entered in Block 20, if different fro neary and identify by block number) IM	m Report)
 DISTRIBUTION STATEMENT (of the ebetract SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfo nested multiplication ABSTRACT (Continue on reverse eide if neces A simple, yet complete an 	entered in Block 20, if different fro meany and identify by block number) IM eary and identify by block number) d detailed descriptic	m Report)
 7. DISTRIBUTION STATEMENT (of the ebstract 7. DISTRIBUTION STATEMENT (of the ebstract 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfonested multiplication 10. ABSTRACT (Continue on reverse eide if neces A simple, yet complete an form for general N is given, 	entered in Block 20, if different fro means and identify by block number) IM eary and identify by block number) d detailed descriptic with the aim of maki	m Report) on of the Fast Fourier Trans- ing the underlying idea quite
 DISTRIBUTION STATEMENT (of the ebstract SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse side if neces FFT = Fast Fourier Transfo nested multiplication ABSTRACT (Continue on reverse side if neces A simple, yet complete an form for general N is given, apparent. To help with this d 	entered in Block 20, if different fro means and identify by block number) TM eary and identify by block number) d detailed descriptic with the aim of maki idactic goal, a simpl	on of the Fast Fourier Trans- ing the underlying idea quite e twist, i.e., a shifting
 DISTRIBUTION STATEMENT (of the ebstract) SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse side if neces) FFT = Fast Fourier Transfonested multiplication ABSTRACT (Continue on reverse side if neces) A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co 	entered in Block 20, if different fro meany and identify by block number) TM eany and identify by block number) d detailed description with the aim of maki idactic goal, a simpl lumns during the calc	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting rulations, is introduced which
 DISTRIBUTION STATEMENT (of the ebstract SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfonested multiplication ABSTRACT (Continue on reverse eide if neces A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co allows to give a simple meaning 	entered in Block 20, if different fro meany and identify by block number) rm d detailed description with the aim of maki idactic goal, a simpl lumns during the calo g to intermediate res	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting culations, is introduced which wults and assures that the
 DISTRIBUTION STATEMENT (of the ebstract) SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse side if necession of the state of the state	entered in Block 20, if different fro meany and identify by block number) m eary and identify by block number) m d detailed descriptic with the aim of maki idactic goal, a simpl lumns during the calc g to intermediate res reordering.	on of the Fast Fourier Trans- ing the underlying idea quite e twist, i.e., a shifting rulations, is introduced which rults and assures that the
 DISTRIBUTION STATEMENT (of the ebstract) SUPPLEMENTARY NOTES KEY WORDS (Continue on reverse eide if neces FFT = Fast Fourier Transfonested multiplication ABSTRACT (Continue on reverse eide if neces A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co allows to give a simple meaning final results need no further in the set of the set o	entered in Block 20, if different fro meany and identify by block number) m eary and identify by block number) m eary and identify by block number) d detailed description with the aim of maki idactic goal, a simpl lumns during the calc g to intermediate res reordering.	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting culations, is introduced which cults and assures that the
 7. DISTRIBUTION STATEMENT (of the ebstract) 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse side if neces) FFT = Fast Fourier Transfonested multiplication 0. ABSTRACT (Continue on reverse side if neces) A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co allows to give a simple meaning final results need no further in the simple in the simple is the simple is the simple in the simple is the simple is the simple in the simple is the simple is the simple is the simple in the simple is the simple is the simple is the simple is the simple in the simple is the sis the simple is the simple is the simple is the simple is	entered in Block 20, if different fro meany and identify by block number) rm eary and identify by block number) d detailed descriptic with the aim of maki idactic goal, a simpl lumns during the calc g to intermediate res reordering.	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting culations, is introduced which ults and assures that the
 7. DISTRIBUTION STATEMENT (of the ebstract 8. SUPPLEMENTARY NOTES 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse side if neces FFT = Fast Fourier Transfonested multiplication 9. ABSTRACT (Continue on reverse side if neces A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co allows to give a simple meaning final results need no further 1 JAN 73 1473 EDITION OF I NOV 65 IS 	entered in Block 20, if different fro entered in Block 20, if different fro entered in Block 20, if different fro entered in the second second second rm eary and identify by block number) m eary and identify by block number) m eary and identify by block number) d detailed description with the aim of maki idactic goal, a simpl lumns during the calc g to intermediate res reordering.	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting culations, is introduced which cults and assures that the NCLASSIFIED
 7. DISTRIBUTION STATEMENT (of the ebstract 8. SUPPLEMENTARY NOTES 9. KEY WORDS (Continue on reverse eide II necest FFT = Fast Fourier Transfonested multiplication 9. ABSTRACT (Continue on reverse eide II necest A simple, yet complete an form for general N is given, apparent. To help with this d of information from rows to co allows to give a simple meaning final results need no further final results need no further for the second statement of the second stateme	entered in Block 20, if different fro meany and identify by block number) m eary and identify by block number) m eary and identify by block number) d detailed descriptic with the aim of maki idactic goal, a simpl lumns during the calc g to intermediate res reordering. OBSOLETE UN SECURITY CLA	on of the Fast Fourier Trans- ing the underlying idea quite the twist, i.e., a shifting rulations, is introduced which rults and assures that the NCLASSIFIED SSIFICATION OF THIS PAGE (When Data Entere

